I.J. Mathematical Sciences and Computing, **2016, 3, 58-70**
Published Online July 2016 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijmsc.2016.03.06

Available online at **http://www.mecs-press.net/ijmsc**

Analysis of Some Software Reliability Growth Models with Learning Effects

Javaid Iqbal

Department of Computer Sciences, University of Kashmir, Srinagar, 190006, India

Abstract

A newly developed software system before its deployment is subjected to vigorous testing so as to minimize the probability of occurrence of failure very soon. Software solutions for safety critical and mission-critical application areas need a much focused level of testing. The testing process is basically carried out to build confidence in the software for its use in real world applications. Thus, reliability of systems is always a matter of concern for us. As we keep on performing the error detection and correction process on our software, the reliability of the system grows. In order to model this growth in the system reliability, many formulations in Software Reliability Growth Models (SRGMs) have been proposed including some based on Non-Homogeneous Poisson Process (NHPP). The role of human learning and experiential pattern gains are being studied and incorporated in such models. The realistic assumptions about human learning behavior and experiential gains of new skill-sets for better detection and correction of faults on software are being incorporated and studied in such models. In this paper, a detailed analysis of some select SRGMs with learning effects is presented based on use of seven data sets. The estimation of parameters and comparative analysis based on goodness of fit using seven data sets are presented. Moreover, model comparisons on the basis of total defects predicted by the select models are also tabulated.

Index Terms: Software Reliability, Software Reliability Growth Model (SRGM), Non-Homogeneous Poisson Process (NHPP), Learning effect, two-type learning effect.

© 2016 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science

1. Introduction

Many software reliability growth models (SRGMs) under the analytical framework of a Non-Homogeneous Poisson Process (NHPP) have been proposed which aim to better model the error-detection and correction processes by trying to incorporate some realistic underlying assumptions. Goel and Okumoto in [1] proposed an exponential SRGM. Yamada and Ohba in [2] proposed delayed S-shaped SRGM while Ohba in [3]

* Corresponding author.
E-mail address: iamjavaid@gmail.com

proposed inflection S-shaped SRGM. Gokhale and Trivedi in [4] proposed an enhanced NHPP model which takes into account the time-dependent failures occurring in debugging process. Debugging process was earlier perceived to be perfect and based on the assumption that that each time an error occurred, the fault that caused it can be removed immediately.

Presently, a more realistic assumption is in place for a debugging process of imperfect type and is based on the assumption that the removal of a fault can introduce some new faults [5, 6]. An insight into imperfect debugging can be found in Obha [3,7], Pham[8], Kapur and Younes [9], Shyur [10] and Chiu and Huang [11]. Other realistic assumptions for running environment, testing/debugging strategies and resource allocation can also affect the reliability growth as discussed in Chiu and Huang [11] and Shyur [10]. Many researchers have used NHPP based SRGMs to capture the reliability growth of a software from the processes of testing and debugging [23-28]. Recently, a unified framework for use of SRGMs with learning process and error generation in imperfect debugging environments has been presented in [12]. Chiu and Huang in [11] proposed a learning-effect-based NHPP model that captures the learning effect gained by testing/ debugging staff from inspection and debugging of code. In [13,14] Chiu proposes an improvement model under time-dependent learning effect. In [5] Iqbal, Ahmad and Quadri propose an SRGM that incorporates two types of learning effects and then incorporate a negligence factor also into the SRGM with two types of learning effect in [6]. They basically indicate that the two types of learning effect are autonomous learning and acquired learning with acquired learning gained after a spell of repeated experience/observation of the testing/debugging process by the tester/debugger resulting in concept formation by the tester/debugger about that particular pattern. Recently in [29, 30] learning based fault detection rates have been incorporated in imperfect debugging models. In this paper, we refine the definition of autonomous learning as the assimilation of know-how by doing (testing) without role of experience and the acquired learning is refined to the definition of learning that stands acquired after a spell of repeated experience/observation of the testing/debugging process by the tester/debugger.

The rest of the paper is organized as: Section II introduces the non-homogeneous Poisson process. Section III discusses how some select learning based models evolved by improvements starting from the learning model proposed by Chiu and Huang in [11], through improvements by Chiu[13] , Iqbal et al [5] introduced the concept of two types of learning in SRGM and later improved it in [6]. This progression in SRGM development is discussed in section III. Section IV discusses parameter estimation. Section V discusses results and presents comparative analysis on the basis of seven data sets listed in the section. This section also presents a comparison of models for total defects predicted using these six out of these seven data Sets. This comparison is presented in six tables. Section VI presents conclusion and is followed by references section. The paper ends with authors' brief profiles.

2. NHPP Modeling Concepts

As an error counting process $\{N(t),\ t\geq 0\}$ with mean $m(t)$ and failure intensity rate $\lambda(t)$ a general NHPP process is written mathematically as:

$$Pr(N(t)=k) = \frac{[m(t)^{k} e^{-m(t)}]}{k!},\ k = 0,1,2,3,\ldots.$$

with mean value function m(t) representing the expected number of errors detected within time (0,t) and mathematically represented as an integral of intensity function between time zero(start) and time t. The conditional software reliability R(s/t) which is the probability that no error is detected within a specific time interval (t, t+s), given that an error has occurred at time t ($t\geq 0$, $s>0$) and is mathematically written as

$$R(s/t) = e^{-[m(t+s)-m(t)]}$$

with limiting value of *R(s/t) ≈ 1* as time approaches to infinity.

3. Developmental Progression of Some Learning-Based Models

Here we present a brief account of the progression of development of some select learning based SRGMs.

A. Chiu and Huang Learning Model [11]:

A learning factor η that arises from inspection of the testing/debugging codes under the assumption that η does not change with time is considered.

Model equation is

$$f(t) = \frac{dF(t)}{dt} = \left(\alpha + \eta F(t)\right)\left(1 - F(t)\right)$$

where autonomous error factor $\alpha > 0$ and learning factor $\eta > 0$.

The explicit solution of $F(t)$ is given by:

$$F(t) = 1 - \frac{1 + \left(\eta/\alpha\right)}{\left(\eta/\alpha\right) + e^{(\alpha+\eta)t}}$$

and

$$f(t) = \frac{(\alpha+\eta)^2 e^{(\alpha+\eta)t}}{\alpha\left(\left(\eta/\alpha\right) + e^{(\alpha+\eta)t}\right)^2}$$

where mean value function $m(t)$ is $m(t) = aF(t)$

$$m(t) = a\left\{1 - \frac{1+\left(\eta/\alpha\right)}{\left(\eta/\alpha\right) + e^{(\alpha+\eta)t}}\right\} \qquad (1)$$

intensity function $\lambda(t) = \frac{d(m(t))}{dt} = af(t)$

and error detection rate is

$$d(t) = \frac{\lambda(t)}{a - m(t)} = (\alpha+\eta)\left(1 - \frac{\eta}{\alpha e^{(\alpha+\eta)t} + \eta}\right)$$

B. Chiu Improvement Model [13,14]:

A learning factor η that arises from inspection of the testing/debugging codes under the assumption that η does not change with time and a negligent factor τ, that arises from negligence on part of testers/developers in correcting errors from learnt patterns previously detected, are considered. Model equation is

$$f(t) = \frac{dF(t)}{dt} = \left(\alpha + \eta F(t) - \tau\right)\left(1 - F(t)\right)$$

The explicit solution of $\boldsymbol{F(t)}$is given by:

$$F(t) = 1 - \frac{1 + \left(\eta/_{(\alpha - \tau)}\right)}{\left(\eta/_{(\alpha - \tau)}\right) + e^{(\alpha+\eta-\tau)t}}$$

and

$$f(t) = \frac{(\alpha + \eta)^2 e^{(\alpha+\eta)t}}{(\alpha - \tau)\left(\left(\eta/_{(\alpha - \tau)}\right) + e^{(\alpha+\eta-\tau)t}\right)^2}$$

Where mean value function$\boldsymbol{m(t)}$is$\boldsymbol{m(t) = aF(t)}$

$$m(t) = a\left\{1 - \frac{1+\left(\eta/_{(\alpha-\tau)}\right)}{\left(\eta/_{(\alpha-\tau)}\right) + e^{(\alpha+\eta-\tau)t}}\right\} \tag{2}$$

intensity function $\boldsymbol{\lambda(t)} = \frac{d(m(t))}{dt} = \boldsymbol{af(t)}$ and error detection rate is

$$d(t) = \frac{\lambda(t)}{a - m(t)} = (\alpha - \tau + \eta)\left(1 - \frac{\eta}{\alpha e^{(\alpha+\eta-\tau)t} + \eta}\right)$$

C. A Two-Type Learning Model [5]:

Two type of learning effect, which are autonomous learning η_1 and acquired learning η_2 which represents experiential gains in learning are considered.

Model equation is

$$f(t) = \frac{dF(t)}{dt} = \left(\eta_1\alpha + \eta_2 F(t)\right)\left(1 - F(t)\right)$$

where autonomous error factor, $\boldsymbol{\alpha > 0}$, type-I learning factor(autonomous learning) $\boldsymbol{\eta_1}$> 0 and type-II learning factor(acquired learning) $\boldsymbol{\eta_2}$> 0. The explicit solution of $\boldsymbol{F(t)}$ is given by

$$F(t) = 1 - \frac{1 + \left(\eta_2/\eta_1\alpha\right)}{\left(\eta_2/\eta_1\alpha\right) + e^{(\eta_1\alpha+\eta_2)t}}$$

and

$$f(t) = \frac{(\eta_1\alpha + \eta_2)^2 e^{(\eta_1\alpha+\eta_2)t}}{\alpha\left(\left(\eta_2/\eta_1\alpha\right) + e^{(\eta_1\alpha+\eta_2)t}\right)^2}$$

The mean value function $\boldsymbol{m(t)}$*is*

$$\boldsymbol{m(t) = aF(t) = a\left\{1 - \frac{1+(\eta_2/\eta_1\alpha)}{(\eta_2/\eta_1\alpha) + e^{(\eta_1\alpha+\eta_2)t}}\right\}}, \quad (3)$$

the intensity function is

$$\boldsymbol{\lambda(t) = \frac{d(m(t))}{dt} = af(t) = a\left\{\frac{(\eta_1\alpha+\eta_2)^2 e^{(\eta_1\alpha+\eta_2)t}}{\alpha\left((\eta_2/\eta_1\alpha) + e^{(\eta_1\alpha+\eta_2)t}\right)^2}\right\}}$$

and variation in error detection rate per error at time $\mathbf{t}$ is given by

$$\boldsymbol{d(t) = \frac{\lambda(t)}{a-m(t)} = (\eta_1\alpha+\eta_2)\left(1 - \frac{\eta_2}{\eta_1\alpha e^{(\eta_1\alpha+\eta_2)t}+\eta_2}\right)}$$

D. ATwo-Type Learning Model with negligence factor[6]

Two types of learning effect, which are autonomous learning η_1 and acquired learning η_2 which represents experiential gains in learning and a negligence factor τ that arises from the negligence on part of testers/developers in correcting errors from learnt patterns previously detected are considered.

The model equation is:

$$\boldsymbol{f(t) = \frac{dF(t)}{dt} = (\eta_1\alpha + \eta_2 F(t) - \tau)(1 - F(t))}$$

where $\boldsymbol{\alpha > 0}$, $\boldsymbol{\eta_1} > 0$ and $\boldsymbol{\eta_2} > 0$.

The explicit solution of $\boldsymbol{F(t)}$ is given by

$$\boldsymbol{F(t) = 1 - \frac{1 + \left(\eta_2/(\eta_1\alpha-\tau)\right)}{\left(\eta_2/(\eta_1\alpha-\tau)\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}}}$$

and

$$\boldsymbol{f(t) = \frac{(\eta_1\alpha+\eta_2-\tau)^2 e^{(\eta_1\alpha+\eta_2-\tau)t}}{\alpha\left(\left(\eta_2/(\eta_1\alpha-\tau)\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}\right)^2}}$$

The mean value function $\boldsymbol{m(t)}$ is

$$\boldsymbol{m(t) = aF(t)}$$

$$\boldsymbol{m(t) = a\left\{1 - \frac{1+\left(\eta_2/(\eta_1\alpha-\tau)\right)}{\left(\eta_2/(\eta_1\alpha-\tau)\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}}\right\}} \quad (4)$$

whereas the intensity function

$$\lambda(t) = \frac{d(m(t))}{dt} = af(t) = a\left\{\frac{(\eta_1\alpha + \eta_2 - \tau)^2 e^{(\eta_1\alpha+\eta_2-\tau)t}}{\alpha\left(\left({\eta_2}/{(\eta_1\alpha-\tau)}\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}\right)^2}\right\}$$

and variation in error detection rate per error at time is given by

$$d(t) = \frac{\lambda(t)}{a-m(t)} = (\eta_1\alpha + \eta_2 - \tau)\left(1 - \frac{\eta_2}{(\eta_1\alpha-\tau)e^{(\eta_1\alpha+\eta_2-\tau)t}+\eta_2}\right)$$

4. Parameter Estimation

Fitting the proposed models to the actual data is done by estimating the model parameters. We have used SPSS to estimate the model parameters by using Regression under Non-linear mode. The estimated parameters are presented in different tables to present a comparative analysis for different listed data sets. The mean value functions represented in equations (1) to (4) are used in estimation of parameters.

The following table-1 presents the datasets by labels and presents the sources of data sets listed.

Table 1. Sources of the Datasets

Label	Reference	Dataset
[1]	Zhang and Pham[16]	Failure Data of Misra System
[2]	Shyur [10]	Failure Data of Misra System
[3]	Hossain and Dahiya[17]	Failure Data of NTDS System
[4]	Pham and Zhang [18]	Failure data of Tandem Software
[5]	Bai, Hu, Xie and Ng [19]	Failure Data of Space program
[6]	Pham[20]	Failure data of real time control system
[7]	Jeske and Zhang [21]	Failure Data of wireless data service system

The following table-2 presents the mean value functions and FDRs of some select models.

Table 2. Model Names and Mean value Function

Goel Okumotto [1]

$$m(t) = a(1- e^{-bt})$$

Chiu and Huang Learning Model [11]

$$m(t) = a\left\{1 - \frac{1 + ({\eta}/{\alpha})}{({\eta}/{\alpha}) + e^{(\alpha+\eta)t}}\right\}$$

2-type Learning Model-1(2TL1) [5]

$$m(t) = a\left\{1 - \frac{1 + ({\eta_2}/{\eta_1\alpha})}{({\eta_2}/{\eta_1\alpha}) + e^{(\eta_1\alpha+\eta_2)t}}\right\}$$

2-type Learning Model-2 (2TL2) [6]

$$m(t) = a\left\{1 - \frac{1 + \left({\eta_2}/{(\eta_1\alpha-\tau)}\right)}{\left({\eta_2}/{(\eta_1\alpha-\tau)}\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}}\right\}$$

The following table-3 presents the values of parameters of select models using seven data sets which are

listed in table 1 using mean value functions listed in table 2.

Table 3. Estimation of Parameters under Seven Data Sets for Select Models

Comparison under dataset[1]	
Model	Parameters
G-O[1]	a= 135.974, b=.138
Chiu[11]	a=135.965, α =.138, η=1.000E-4
2TL1[5]	a=135.965, α =49.216, η1 =.003, η2 =1.000E-4
2TL2[6]	a=135.974, α =2.611, η1 =2.566, η2 =.001, τ =6.562
Comparison under dataset[2]	
Model	Parameters
G-O[1]	a= 218.159, b=.041
Chiu[11]	a=215.706, α =.042, η=.001
2TL1[5]	a=215.706, α =56.69, η1 =.001, η2 =.001
2TL2[6]	a=210.134, α =.094, η1 =.442, η2 =1.000E-5, τ =7.007E-5
Comparison under dataset[3]	
Model	Parameters
G-O[1]	a= 33.6, b=.063
Chiu[11]	a=24.821, α =.024, η=.343
2TL1[5]	a=24.821, α =.056, η1 =.424, η2 =.343
2TL2[6]	a=24.821, α =.217, η1 =.658, η2 =.343, τ =.119
Comparison under dataset[4]	
Model	Parameters
G-O[1]	a= 133.761, b=.015
Chiu[11]	a=133.496, α =.146, η=.001
2TL1[5]	a=133.496, α =153.843, η1 =.001, η2 =.001
2TL2[6]	a=133.496, α =.002, η1 =909.569, η2 =.001, τ =1.748
Comparison under dataset[5]	
Model	Parameters
G-O[1]	a= 18.257, b=.397
Chiu[11]	a=18.254, α =.397, η=.001
2TL1[5]	a=18.254, α =.049, η1 =8.151, η2 =.001
2TL2[6]	a=18.257, α =23.658, η1 =.665, η2 =1.000E-5, τ =15.341
Comparison under dataset[6]	
Model	Parameters
G-O[1]	a= 124.44, b=.051
Chiu[11]	a=124.171, α =.051, η=.001
2TL1[5]	a=124.171, α =.001, η1 =88.486, η2 =.001
2TL2[6]	a=124.437, α =.163, η1 =5.874, η2 =1.000E-5, τ =.909
Comparison under dataset[7]	
Model	Parameters
G-O[1]	a= 23.092, b=.559
Chiu[11]	a=22.252, α =.493, η=.332
2TL1[5]	a=22.252, α =.211, η1 =.2.338, η2 =.332
2TL2[6]	a=22.252, α =9.048, η1 =.195, η2 =.332, τ =1.272

5. Results and Comparative Analysis

There are many comparison criteria as defined in [22] wherein the authors have presented analysis and ranking of software reliability models based on weighted criteria. The comparison criteria used is R^2 measure also called as coefficient of multiple determinations (R^2) which is usually used to depict the goodness-of-fit and is expressible as: [15]

$$\boldsymbol{R^2 = 1 - \frac{(ResidualSumofSquares)}{(CorrectedSumofSquares)}}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(m_i - m(t_i))^2}{\sum_{i=1}^{n}(m_i - \sum_{k=1}^{n} {m_k}/{n})^2}$$

R^2 represents a measure of the percentage of the total variation about the mean for the fitted curve. It lies in the range of 0 to 1, with a larger R^2 value indicating a better representation of variation about the mean of the data set by the model equation. However, a smaller R^2value indicates that the model equation fails to represent the variations in the data set. Obviously, a near-one value of R^2 is highly desirable [15]. A comparative analysis of some select models using R^2 measure is presented using seven datasets and model comparisons on the basis of total defects predicted by the select models are also tabulated.

The following table-4 presents the results of goodness-of-fit under R^2comparison criteria

Table 4. Goodness-of-fit under R2comparison Criteria

Comparison under R-sq for given datasets				
Dataset	G-O [1]	Chiu[11]	2TL1[5]	2TL2 [6]
[1]	.966	.966	.966	.966
[2]	.989	.989	.989	.989
[3]	.919	.992	.992	.992
[4]	.990	.990	.990	.990
[5]	.934	.934	.934	.934
[6]	.978	.977	.977	.978
[7]	.987	.989	.989	.989

The following tables 5-11 present the comparison of models for total defects predicted under listed data sets

Table 5. Comparison of Models for Total Defects Predicted using Data Set [1]

Defects	GO[1]	Chiu[11]	2TL1[5]	2TL2[6]
27	17.55773	17.55748	17.55509	17.53115
43	32.84831	32.84799	32.8448	32.81288
54	46.1645	46.1642	46.16116	46.13079
64	57.76122	57.76101	57.75864	57.73498
75	67.86052	67.86041	67.85894	67.84422
82	76.65575	76.65574	76.6552	76.6498
84	84.31529	84.31537	84.31568	84.31883
89	90.98579	90.98595	90.98697	90.99726
92	96.79496	96.79517	96.79674	96.81247
93	101.854	101.8543	101.8562	101.8756
97	106.2598	106.2601	106.2622	106.2836
104	110.0967	110.097	110.0992	110.1209
106	113.4382	113.4384	113.4405	113.4613
111	116.3482	116.3484	116.3503	116.369
116	118.8824	118.8826	118.8842	118.8999
122	121.0895	121.0896	121.0908	121.1028
122	123.0115	123.0116	123.0123	123.0201
127	124.6853	124.6853	124.6857	124.6887
128	126.143	126.143	126.1428	126.141
129	127.4125	127.4124	127.4117	127.4049
131	128.5181	128.5179	128.5167	128.5048
132	129.4809	129.4807	129.479	129.462
134	130.3194	130.3191	130.3169	130.2951
135	131.0496	131.0493	131.0466	131.02
136	131.6855	131.6851	131.682	131.6509

Table 6.Comparison of models for Total Defects Predicted using Data Set [2]

Defects	GO[1]	Chiu[11]	2TL1[5]	2TL2[6]
13	9.389554	9.377357	9.377357	5.560235
20	12.8108	12.79549	12.79549	9.389434
26	18.49874	18.47969	18.47969	12.81065
31	23.54737	23.52641	23.52641	18.49855
34	28.78153	28.75984	28.75984	23.54717
36	34.40701	34.38584	34.38584	28.78131
41	39.90221	39.88266	39.88266	34.4068
45	46.54956	46.53319	46.53319	39.90202
47	51.53715	51.52387	51.52387	46.54939
51	55.91801	55.9078	55.9078	51.53702
58	60.19674	60.18975	60.18975	55.91791
58	62.52843	62.52325	62.52325	60.19667
63	65.9616	65.9591	65.9591	62.52838
66	68.42484	68.42425	68.42425	65.96157
69	72.57233	72.57486	72.57486	68.42483
72	76.19502	76.20011	76.20011	72.57235
76	79.81904	79.82643	79.82643	76.19507
86	85.34595	85.35619	85.35619	79.81911
89	86.71047	86.72127	86.72127	85.34605
90	87.36045	87.37149	87.37149	86.71058
92	90.29866	90.31055	90.31055	87.36056
96	92.8607	92.87302	92.87302	90.29878
101	96.12987	96.14225	96.14225	92.86082
101	97.43348	97.44571	97.44571	96.12999

Table 7. Comparison of Models for Total Defects Predicted using Data Set [3]

Defects	GO[1]	Chiu[11]	2TL1[5]	2TL2[6]
1	1.851063	0.616585	0.616584	0.616585
2	4.161494	1.743671	1.743667	1.743671
3	6.131381	3.151695	3.151688	3.151694
4	6.814544	3.763111	3.763104	3.76311
5	7.969451	4.964748	4.96474	4.964748
6	8.290182	5.338404	5.338395	5.338403
7	9.07456	6.328614	6.328605	6.328613
8	10.27932	8.061859	8.061851	8.061859
9	11.00205	9.219136	9.219129	9.219136
10	11.9764	10.8984	10.89839	10.8984
11	12.11212	11.1413	11.14129	11.1413
12	12.90874	12.59865	12.59865	12.59865
13	13.03861	12.84003	12.84002	12.84003
14	14.17135	14.95556	14.95555	14.95556
15	14.65455	15.8461	15.8461	15.8461
16	14.77346	16.06253	16.06253	16.06253
17	15.12573	16.69539	16.69539	16.69539
18	15.47142	17.30184	17.30184	17.30184
19	16.14349	18.42877	18.42877	18.42877
20	16.25306	18.60495	18.60495	18.60495
21	17.41382	20.3175	20.3175	20.3175
22	20.45052	23.29559	23.29559	23.29559
23	21.01748	23.62514	23.62514	23.62514
24	26.5052	24.77606	24.77605	24.77606
25	26.59398	24.7792	24.77919	24.7792

Table 8. Comparison of Models for Total Defects Predicted using Data Set [4]

Defects	GO[1]	Chiu[11]	2TL1[5]	2TL2[6]
16	9.770512	9.763303	9.763301	9.763301
24	17.64551	17.6352	17.6352	17.6352
27	25.22674	25.21546	25.21546	25.21546
33	32.32782	32.31718	32.31717	32.31717
41	40.80269	40.79441	40.7944	40.7944
49	48.20764	48.20246	48.20246	48.20246
54	55.01122	55.00938	55.00937	55.00937
58	63.66981	63.67229	63.67229	63.67229
69	71.36726	71.37296	71.37295	71.37295
75	76.64714	76.65427	76.65426	76.65426
81	82.3215	82.32902	82.32902	82.32902
86	86.25277	86.25962	86.25962	86.25962
90	88.97659	88.98241	88.98241	88.98241
93	91.26569	91.27022	91.27023	91.27023
96	93.43779	93.44069	93.4407	93.4407
98	95.49886	95.49981	95.49982	95.49982
99	97.45458	97.45328	97.45328	97.45328
100	99.31033	99.30648	99.30649	99.30649
100	101.0712	101.0646	101.0646	101.0646
100	102.7421	102.7324	102.7324	102.7324

Table 9. Comparison of Models for Total Defects Predicted using Data Set [5]

Defects	GO[1]	Chiu[11]	2TL1[5]	2TL2[6]
1	0.144261	0.144189	0.144189	0.144262
2	0.429373	0.429168	0.429166	0.429375
3	1.054572	1.05412	1.054115	1.054578
4	1.854199	1.853515	1.853508	1.85421
5	2.740242	2.739408	2.739397	2.740259
6	3.752057	3.75118	3.751166	3.752081
7	5.1214	5.120669	5.120651	5.121434
8	6.733016	6.732716	6.732696	6.733063
9	8.266495	8.266813	8.266793	8.266553
10	9.998512	9.999685	9.999666	9.998582
11	11.64348	11.6455	11.64548	11.64356
12	13.00265	13.00525	13.00524	13.00272
13	14.37015	14.37307	14.37307	14.37022
14	15.72448	15.72717	15.72718	15.72453
15	16.7745	16.77626	16.77629	16.77452
16	17.44575	17.44627	17.4463	17.44574
17	17.9312	17.93013	17.93018	17.93116
18	18.15346	18.1512	18.15126	18.1534
19	18.2279	18.22505	18.22511	18.22783
20	18.24948	18.2464	18.24646	18.2494
21	18.25548	18.25232	18.25238	18.2554

Table-10: Comparison of models for total defects predicted using data Set [6]:

We skip analysis for the lengthy real time control data set. However R^2 analysis is presented for this dataset also.

Table 10. Comparison of Models for Total Defects Predicted using Data Set [7]

Defects	GO[1]	Chiu[11]	2TL1[5]	2TL2[6]
1	1.66715	1.451695	1.451693	1.451695
2	2.843703	2.52046	2.520456	2.52046
4	3.955647	3.56411	3.564105	3.56411
5	4.305553	3.899213	3.899208	3.899213
6	6.929299	6.510605	6.510599	6.510605
7	7.233704	6.824362	6.824356	6.824362
9	8.378618	8.022904	8.022898	8.022904
10	10.67193	10.50022	10.50022	10.50022
11	12.42447	12.44267	12.44266	12.44267
12	13.20017	13.30905	13.30905	13.30905
14	13.38647	13.51728	13.51727	13.51728
15	13.74862	13.92189	13.92189	13.92189
16	15.92825	16.33299	16.33299	16.33299
18	16.47513	16.92416	16.92416	16.92416
20	19.45476	19.91598	19.91598	19.91598
21	21.69196	21.66205	21.66205	21.66205
22	21.74798	21.6962	21.6962	21.6962
22	21.9273	21.80123	21.80123	21.80123
22	22.46186	22.06916	22.06916	22.06916

6. Conclusion

In this paper, a detailed analysis of some select SRGMs with learning effects is presented, the developmental progression is shown. Seven data sets have been used for this detailed analysis and parameter estimation is also presented based on seven data sets. The parameters estimated are competing with other famous models. R^2comparison criteria shows fairly good values to validate the models. Finally, model comparisons on the basis of total defects predicted by the select models are also presented in tables.

References

[1] A. L. Goel and K. Okumoto, “Time-dependent error-detection rate model for software and other performance measures,” IEEE Transactions on Reliability, vol. 28, pp. 206–211, 1979.

[2] S. Yamada and H. Ohba “S-shaped software reliability modeling for software error detection”, 1983, IEEE Trans Reliab; 32:475–84.

[3] M. Ohba, “Inflexion S-shaped software reliability growth models”, Stochastic Models in Reliability Theory (S. Osaki, Y. Hatoyama, Eds), 1984, pp 144 - 162; Springer- Verlag Merlin.

[4] S.S. Gokhale, and K.S.Trivedi “A time/structure based software reliability model” Ann Software Eng; 1999, 8:85–121.

[5] J. Iqbal, N. Ahmad, and S.M.K. Quadri, “A Software Reliability Growth Model with Two types of Learning”, Proceedings of the 1st IEEE International Conference on Machine Intelligence Research and Advancement, SMVDU, Jammu, India, pp. 498–503, 2013.

[6] J. Iqbal, N. Ahmad, and S.M.K. Quadri, "A software reliability growth model with two types of learning and a negligence factor," Image Information Processing (ICIIP), 2013 IEEE Second International Conference on , vol., no., pp.678,683, 9-11 Dec. 2013.

[7] M. Ohba, “Software reliability analysis models,” IBM Journal of Research Development, vol. 28, pp. 428–443, 1984.

[8] H. Pham, “Software reliability assessment: Imperfect debugging and multiple failure types in software development”, 1993; EG&G-RAAM-10737, Idaho National Engineering Laboratory.

[9] P.K. Kapur, and S. Younes, “Modeling an imperfect debugging phenomenon in software reliability”, Microelectronics and Reliability, 1996, Vol. 36, pp. 645-50.

[10] H.J. Shyur, “A stochastic software reliability model with imperfect debugging and change-point”, J Syst Software, 2003, 66:135–41.

[11] K.C. Chiu, Y.S. Huang, and T.Z. Lee, “A study of software reliability growth from the perspective of learning effects”, Reliability Engineering and System Safety, 2008, 93: 1410-1421.

[12] P. K. Kapur, H. Pham, S. Anand, K. Yadav, “A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation”, IEEE Transactions on Reliability 60 (1) , 331–340, 2011.

[13] K.C. Chiu, "An improved model of software reliability growth under time-dependent learning effects," Quality and Reliability (ICQR), 2011 IEEE International Conference on, vol., no., pp.191, 194, 14-17 Sept. 2011.

[14] K.C. Chiu, “A discussion of software reliability growth models with time-varying learning effects”, American Journal of Software engineering and applications. Vol. 2, No. 3, 2013, pp 92-104. doi: 10.11648/j.ajsea.20130203.12

[15] V. B. Singh, P. K. Kapur, and Mashaallah Basirzadeh. "Open Source Software Reliability Growth Model by Considering Change-Point." BVICAM's International Journal of Information Technology 4.

[16] Zhang X, Pham H. A software cost model with warranty cost, error removal times and risk costs. IIE Trans 1998;30:1135–42

[17] Hossain SA, Dahiya RC. Estimating the parameters of a non homogeneous Poisson-process model for software reliability. IEEETrans Reliab 1993; 42:604–12.

[18] Pham H, Zhang X. NHPP software reliability and cost models with testing coverage. Eur J Oper Res 2003;145:445–54

[19] Bai CG, Hu QP, Xie M, Ng SH. Software failure prediction based on a Markov Bayesian network model. J Syst Software 2005;74:275–82

[20] Pham H. Software reliability and cost models- perspectives, comparison, and practice. Eur J Oper Res 2003;149:475–89

[21] Jeske DR, Zhang X. Some successful approaches to software reliability modeling in industry. J Syst Software 2005;74:85–99

[22] Anjum Mohd., Md. Asraful Haque, Nesar Ahmad. Analysis and Ranking of Software Reliability Models Based on Weighted Criteria Value, I.J. Information Technology and Computer Science, 2013, 02, pp.1-14, DOI: 10.5815/ijitcs.2013.02.01

[23] N. Ahmad, M.G.M. Khan, and L.S. Rafi, “Analysis of an Inflection S-shaped Software Reliability Model Considering Log-logistic Testing-Effort and Imperfect Debugging”, International Journal of Computer Science and Network Security, 2011, Vol. 11 (1), pp. 161 – 171.

[24] N. Ahmad, M.G.M. Khan, and L.S. Rafi, “A Study of Testing-Effort Dependent Inflection S-Shaped Software Reliability Growth Models with Imperfect Debugging”, International Journal of Quality and Reliability Management, 2010, Vol. 27 (1), pp. 89 – 110.

[25] P.K. Kapur, P.K. and S. Younes, “Modeling an imperfect debugging phenomenon in software reliability”, Microelectronics and Reliability, 1996, Vol. 36, pp. 645-50.

[26] N. Ahmad, M.G.M. Khan, and L.S. Rafi, “Inflection S-shaped software reliability growth models with testing-effort functions,” Proceedings of the VI International Symposium on Optimization and Statistics, Aligarh Muslim University, Aligarh, India, 29-31 December, 2008.

[27] N. Ahmad, M.U. Bokhari, S.M.K. Quadri, and M.G.M. Khan, “The exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy: a performance analysis”, International Journal of Quality & Reliability Management, 2008, Vol. 25 No. 2, pp. 211-35.

[28] N. Ahmad, M.G.M. Khan, and L.S. Rafi, “Software Reliability Modeling Incorporating Log-Logistic Testing-Effort with Imperfect Debugging”, in Proceedings of the International Conference on Modeling, Optimization and Computing (ICMOC-2010), Durgapur, India, Published by American Institute of

Physics, 2010, pp. 651 – 657.
[29] J. Iqbal, S.M.K. Quadri, and N. Ahmad, "Software Reliability Modeling with Learning-Factor Based Fault-Detection Rate", Proceedings of the 1st International Conference on Recent Trends in Computer Science & Engineering" 8-9th February, 2014, Patna Bihar India (Narosa Publishing House)
[30] J. Iqbal, S.M.K. Quadri, and N. Ahmad, "An Imperfect- Debugging Model with Learning-Factor Based Fault-Detection Rate", Proceedings of the 8th International Conference on "Computing for Sustainable Global Development" (INDIACom-2014), Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA), 5th– 7thMarch 2014.

Authors' Profiles

Javaid Iqbal received his B.Sc in Mathematics from the University of Kashmir, and Masters in Computer Applications from the same university in 2004. He completed his Ph.D from the University of Kashmir in 2014. His research interests are Software engineering, Software reliability engineering and reliability modeling, and mathematical modeling of dynamic systems. He is a member of ACM, CSI and IETE.

How to cite this paper: Javaid Iqbal,"Analysis of Some Software Reliability Growth Models with Learning Effects", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.2, No.3, pp.58-70, 2016.DOI: 10.5815/ijmsc.2016.03.06